# College Library Management System (LMS)

## Technical & Functional Documentation

Project Status: In Progress

Last Updated: January 14, 2026

## 1. Introduction

*This document outlines the requirements for a College Library Management System designed as an internal administrative tool. The system allows Staff and Admins to manage book inventories, circulation, and library members. Note: This system is for internal use only; students do not have direct access.*

## 2. System Architecture

The system is built as a **Desktop Application** using the **MERN Stack** wrapped in **Electron.js**, utilizing a **Hybrid Cloud-Sync Architecture** to support multiple devices and users.

* **Client (Desktop App):**
  + **Framework:** Electron.js (Cross-platform desktop container).
  + **Local Persistence (Native Store):** Local MongoDB Instance (**mongod**) that stores data natively on the device. This ensures zero-latency interactions and limited offline capability.
  + **UI Library:** React.js (Frontend user interface).
  + **Usage:** Installed on multiple devices (Library Counters, Admin Cabins) acting as synchronized nodes.
* **Cloud Backend (Central Hub):**
  + **Database:** **MongoDB Atlas** (Cloud-hosted Cluster). Acts as the "Single Source of Truth". *The connection URI is manually configured by the Admin in the application settings.*
  + **API Server:** **Integrated App Server**. The Node.js/Express backend runs locally within the Electron application instance (or as a sidecar process). It connects directly to the provided MongoDB Cloud URI. **No external server hosting is required.**
  + **Sync Engine:** Utilizes **Socket.io** or **MongoDB Change Streams** to broadcast updates.
* **Synchronization Flow (Auto-Update):**
  1. **Local Write:** When a Staff issues a book, the transaction is stored immediately in the **Local Native Store**.
  2. **Push to Cloud:** The application automatically pushes this change to **MongoDB Atlas** over the internet via the Integrated App Server.
  3. **Broadcast (Pull):** MongoDB Atlas triggers an update event. All other connected devices (e.g., Admin's PC) receive this update and modify their local data instantly.
  4. **Connectivity:** Active Internet connection is required for synchronization. If the internet is lost, data is queued locally and **auto-synced** once connectivity is restored.
* **Conflict Resolution Strategy:**
  + **Policy:** **Last Write Wins (LWW)**.
  + **Scenario:** If multiple staff members edit the same record (e.g., Book Metadata) while offline or simultaneously on different devices.
  + **Resolution Logic:** The update that reaches the Cloud Database (MongoDB Atlas) with the latest timestamp overrides previous versions.
  + **Audit Trail:** To ensure accountability, the system automatically creates an entry in the **Audit\_Log** for the overwritten change (classifying it as a CONFLICT\_RESOLVED event), storing the previous value in the metadata field for review.

## 3. User Roles & Permissions

The system distinguishes between active system users (Admin, Staff) and passive entities (Students).

### 3.1. Admin

* **Description:** Users with System-level authority.
* **Permissions:**
  + **Inherited Permissions:** Includes all access rights granted to **Staff** (can perform circulation, cataloging, etc.).
  + **System Configuration:** Configure global settings, library hours, and holidays.
  + **Role & Permission Management:** Create and manage user roles, assign permissions, and oversee staff access.
  + **Policy Definition:** Define and update fine calculation rules, borrowing limits, and renewal policies.
  + **Master Data Control:** Full control over core data (Departments, Vendors).
  + **Reports & Audits:** Access comprehensive system logs, financial reports, and inventory audits.

### 3.2. Staff

* **Description:** Library Assistants / Counter Staff.
* **Permissions:**
  + **Catalog Management:** Add, update, or remove books/periodicals.
  + **Circulation:** Issue and return books for students (Borrowers).
  + **Fine Collection:** Collect fines and clear dues on behalf of students.
  + **Member Management:** View student details, history, and block/unblock borrowing privileges.

### 3.3. Student (Passive Entity)

* **Description:** College students (Borrowers). **They do not have login access to the system.**
* **Role in System:**
  + Identification via Register Number / ID Card.
  + Borrowing and returning books via the Circulation Desk.
  + Paying fines via Staff.

## 4. Functional Modules

### 4.1. Authentication Module

* **Features:**
  + **Single Login Strategy:**
    - **Admin & Staff:** Login via Email + Password.
  + **Access Control:**
    - Strict Role-Based Access Control (RBAC) preventing unauthorized access to Admin modules.
  + **Session Management:** Secure token-based authentication (JWT) stored securely within the Electron app (e.g., electron-store or secure HTTP-only cookies).

### 4.2. Catalog Management Module (Staff/Admin)

* **Features:**
  + Add new books (ISBN, Title, Author, Publisher, Edition).
  + Manage copies (Assigning unique **Accession Numbers** in the format {ISBN}-{Sequence} to separate physical copies, e.g., 9781234567890-001).
  + **Department Assignment:** Tag books to specific departments (e.g., Computer Science, Mechanical) for organized retrieval.
  + Stock audit and status tracking (Available, Issued, Lost, Damaged).

### 4.3. Circulation Module (Staff Only)

* **Features:**
  + **Issue Book:** Staff enters Student ID and **Book Accession Number** to link them; system sets due date.
  + **Return Book:** Staff enters **Book Accession Number**; system auto-calculates fines if overdue.
  + **Renewals:** Staff extends due dates upon student request (in-person).

### 4.4. Fine & Alert Module

* **Features:**
  + Auto-calculation of late fees based on (Return Date - Due Date) \* Daily Rate.
  + **Staff Notifications:** Alerts on the dashboard for overdue items and blocked students.

### 4.5. UI/UX & Personalization Module

* **Features:**
  + **Internationalization (i18n):**
    - **Languages:** Support for **English** (Primary) and **Kannada**.
    - **Mechanism:** Instant toggle in the application header. UI labels and messages loaded dynamically from translation files (e.g., en.json, kn.json).
  + **Theme Management:**
    - **Visual Style:** **Glassmorphism**.
      * **Core Aesthetic:** Frosted glass effects (background blur), multi-layered transparency, and soft, vibrant gradients.
      * **Components:** Floating "glass" cards, thin white borders, and subtle shadows to establish depth.
    - **Modes:** Built-in **Light Mode** (default) and **Dark Mode** (high contrast).
    - **Persistence:** Application remembers the user's preference across sessions.
  + **Accessibility & Display:**
    - **Font Sizing:** Global settings to increase/decrease text size (Small, Medium, Large, Extra Large).
    - **Responsive Layout:** UI components adapt to font scaling without breaking layout (using relative units like rem/em).

### 4.6. Notification & Broadcast Module

* **Features:**
  + **Bulk Communication:** integrated email service (e.g., SMTP, SendGrid, or AWS SES) to send mass notifications.
  + **Dynamic Targeting:**
    - **Global:** Send to All Students.
    - **Departmental:** Filter recipients by selecting one or multiple departments (e.g., "Computer Science" + "Electronics").
    - **Individual:** Search and select specific students by Name or Register Number.
    - **Behavioral/Status-Based:**
      * **Overdue Students:** Auto-targets students who currently have overdue items.
      * **Issued Students:** Targets students currently holding library assets.
  + **Content Management:** Rich text editor for composing message bodies with support for basic formatting (bold, lists, links).
  + **Audit Log:** History of sent broadcasts, including timestamp, sender, target group, and success/failure count.

### 4.7. Email Service Management Module

* **Features:**
  + **Global Control:** Master switch to Enable/Disable all system emails instantly (useful during maintenance).
  + **Provider Flexibility:** Support for standard **SMTP** (Gmail, Outlook, Custom Domain) and **Cloud APIs** (SendGrid, AWS SES) for high-volume delivery.
  + **Event Triggers:** Granular control to toggle specific email notifications (e.g., Enable "Issue Receipts" but Disable "Welcome Emails").
  + **Test Connectivity:** Built-in tool to validate credentials by sending test emails before saving configuration.
  + **Template Engine:**
    - **Visual Identity:** Emails must use **Responsive HTML Templates** that mirror the application's visual theme (Logo, Brand Colors, clean typography).
    - **Dynamic Theming:** Templates should attempt to respect user's system theme preference where possible, or default to a clean, professional "White Paper" style for high legibility.
    - **Components:** Standardized Header (Logo), Body (Card-style container), and Footer (Contact Info, Disclaimer).

## 5. Data Entities (MongoDB Schemas)

### 5.1. Student Schema (Borrower Record)

* \_id (PK): ObjectId (Auto-generated).
* register\_number: String (Unique index; Not Null).
* full\_name: String (Not Null).
* dept\_id: ObjectId referencing Department Collection (Not Null).
* semester: Number (Enum: 1, 2, 3, 4, 5, 6).
* email: String (Not Null).
* phone: String.
* dob: Date (Not Null).
* address: String (Optional).
* status: String Enum (Active, Blocked, Alumni).

### 5.2. Staff Schema

* \_id (PK): ObjectId (Auto-generated).
* name: String (Not Null).
* email: String (Unique index).
* phone: String.
* designation: String (e.g., "Assistant Librarian", "Counter Staff").
* access\_permissions: Array of Strings (e.g., ["issue\_books", "manage\_catalog"]).
* password\_hash: String.
* status: String Enum (Active, Disabled).
* created\_at: Date.
* last\_login: Date.

### 5.3. Admin Schema

* \_id (PK): ObjectId (Auto-generated).
* name: String.
* email: String (Unique index).
* phone: String.
* password\_hash: String.
* status: String Enum (Active, Disabled).
* created\_at: Date.
* last\_login: Date.

### 5.4. Book Schema (Bibliographic Record)

* \_id (PK): ObjectId.
* isbn: String (Unique index). *Note: Stores standard ISBN or auto-generated "AG-" IDs.*
* title: String.
* author: String.
* publisher: String.
* dept\_id: ObjectId referencing Department Collection.
* price: Number (Decimal).
* cover\_image\_url: String.
* ebook\_link: String (Optional).
* total\_copies: Number.

### 5.5. Book\_Copy Schema (Physical Item)

* \_id (PK): ObjectId.
* accession\_number: String (Unique index; Format: {ISBN}-{Sequence}, e.g., 986985868565-001).
* book\_id: ObjectId referencing Book Schema.
* status: String Enum (Available, Issued, Lost, Maintenance).
* location: String (Shelf Number).

### 5.6. Transaction Schema (Loans)

* \_id (PK): ObjectId.
* session\_txn\_id: String (Audit Ref: TXN-DDMMYYYY-XXXX). *Groups multiple book issues into a single checkout session. Example: TXN-14012026-A92B.*
* student\_id: ObjectId referencing Student Schema.
* copy\_id: ObjectId referencing Book\_Copy Schema.
* issued\_by: ObjectId referencing Staff Schema.
* issue\_date: Date.
* due\_date: Date.
* return\_date: Date (Nullable).
* status: String Enum (Active, Returned, Overdue).

### 5.7. Fine Schema

* \_id (PK): ObjectId.
* receipt\_number: String (Unique, Auto-generated, e.g., "F-2026-0001").
* transaction\_id: ObjectId referencing Transaction Schema.
* student\_id: ObjectId referencing Student Schema (Indexed for search).
* amount: Number (Decimal).
* is\_paid: Boolean.
* payment\_date: Date.
* collected\_by: ObjectId referencing Staff Schema.

### 5.8. Department Schema

* \_id (PK): ObjectId.
* name: String (Unique, e.g., "Computer Science Engineering").
* code: String (Unique, e.g., "CSE").
* description: String.

### 5.9. Broadcast\_Log Schema (Notification History)

* \_id (PK): ObjectId.
* sender\_id: ObjectId referencing Staff/Admin Schema.
* sent\_at: Date.
* subject: String.
* message\_body: String (HTML/Text).
* target\_group: String (e.g., "All Students", "Dept: CSE").
* recipient\_count: Number.
* status: String Enum (Sent, Failed).

### 5.10. Audit\_Log Schema (System-Wide Events)

* \_id (PK): ObjectId.
* timestamp: Date (Indexed).
* actor\_id: ObjectId referencing Staff/Admin.
* actor\_role: String (Staff/Admin).
* action\_type: String Enum (LOGIN, CREATE, UPDATE, DELETE, EXPORT, SETTINGS\_CHANGE).
* module: String (e.g., "Catalog", "User Management", "Circulation").
* description: String (e.g., "Deleted Book: Intro to AI").
* ip\_address: String.
* metadata: Object (Optional: Stores 'Before' and 'After' values for critical changes).

### 5.11. Email\_Config Schema (System Settings)

* \_id (PK): ObjectId (Singleton Document).
* service\_status: Boolean (Master Enable/Disable).
* provider: String Enum (SMTP, SENDGRID, AWS\_SES).
* smtp\_settings:
  + host: String.
  + port: Number.
  + secure: Boolean.
  + auth\_user: String.
  + auth\_pass: String (Encrypted).
* cloud\_settings:
  + api\_key: String (Encrypted).
  + region: String (For AWS).
* sender\_identity:
  + from\_email: String.
  + from\_name: String.
* triggers: Object (Boolean flags for events: issue\_receipt, return\_receipt, renewal\_receipt, overdue\_alert, broadcast\_msg).

### 5.12. Policy\_Config Schema (Versioning & Rules)

* \_id (PK): ObjectId.
* policy\_version: Number (Incrementing, e.g., v1.0, v1.1).
* active\_from: Date.
* profiles:
  + student\_active: Object (Max Books, Days, Renewals, Fine Threshold).
  + student\_alumni: Object (Stricter limits).
  + staff: Object (Higher limits, Auto-renew status).
* financials:
  + fine\_rate\_daily: Number.
  + max\_fine\_cap: Number.
  + damage\_presets: Array of Objects ([{ label: "Torn Page", amount: 50 }, ...]).
* holidays:
  + weekly\_off: Array (e.g., ["Sunday"]).
  + exclude\_fines\_on\_holidays: Boolean.
* security:
  + max\_login\_attempts: Number.
  + require\_admin\_auth\_for: Array (e.g., ["BULK\_DELETE", "POLICY\_CHANGE"]).

## 6. Data Flow

### 6.1. Staff/Admin Login Flow

1. **Input:** User opens Electron App -> Enters Email + Password.
2. **Process:**
   * Client sends POST request to Express Server.
   * Server verifies credentials in MongoDB.
   * Returns JWT Token.
3. **Output:** Electron App stores Token; Redirects to React Dashboard.

### 6.2. Book Issue Flow (Circulation Desk)

1. **Trigger:** Student requests book at counter.
2. **Staff Action:** Enters Student Register Number + Enters Book Accession Number.
3. **System Validation:**
   * Is Student active?
   * Is Book status 'Available'?
   * Has Student reached max borrow limit?
4. **Data Action:**
   * Create Transaction document.
   * Update Book\_Copy status to 'Issued'.
5. **Output:** Success toast notification; Transaction logged; **Email Receipt sent to Student.**

### 6.3. Book Return Flow

1. **Staff Action:** Enters Book Accession Number.
2. **System Process:**
   * Query active Transaction for this copy.
   * Checks: Current Date > Due Date?
   * If Yes: Calculate Fine and create Fine document.
3. **Data Action:**
   * Update Transaction with return\_date.
   * Update Book\_Copy status to 'Available'.
4. **Output:** Show "Return Successful" modal; **Email Receipt sent to Student.**

## 7. Non-Functional Requirements

* **Security:** Passwords hashed with bcrypt. API endpoints protected via JWT middleware.
* **Performance:** React Virtualization for large lists. Search processing < 100ms.
* **Offline Capability:** Native Store caching for offline resilience.
* **Scalability:** MongoDB clustered architecture.
* **Localization:** UI support for English & Kannada (Unicode).
* **Accessibility:** WCAG 2.1 compliance (Contrast, Scaling).
* **User Feedback & States:**
  + **Global Loading:** All async actions (API calls) must trigger a **Spinner Overlay** to prevent double-submissions.
  + **Live Progress Tracking:** Long-running processes (Bulk CSV Import, Large PDF Export, Database Restore) must display a **Real-Time Progress Bar** showing percentage complete (e.g., "Processing row 450/1000 - 45%").
* **Data Integrity & Validation (Referential Integrity):**
  + **Strict Dependency Checks:** The system must enforce server-side checks before deletion.
  + **Rules:**
    - **Cannot Delete Department** if linked to any Student or Book.
    - **Cannot Delete Student** if they have active Transactions (Unreturned books) or Unpaid Fines.
    - **Cannot Delete Book** if any copy is currently Issued.
    - **Cannot Delete Staff** if they are recorded as the issuer/collector on any Transaction or Fine (Use "Disable/Soft Delete" instead).

## 8. Interface Specifications (Page-by-Page)

This section details the layout, functionality, and visual hierarchy for each screen, applying the **Glassmorphism** design system.

### 8.0. General Layout & Responsiveness (Global Rule)

* **Auto-Sizing & Fluidity:** Every interface must automatically adjust to fit the user's screen size, ranging from compact laptop screens to large desktop monitors.
* **Responsive Glass Panels:** The central glass containers and cards must use relative width/height (percentages, vh, vw) rather than fixed pixels to ensure they resize proportionally.
* **Layout Adaptation:**
  + On **Wide Screens:** Content should expand horizontally (e.g., 3-column grids).
  + On **Narrow Screens:** Content should stack vertically or switch to a scrollable view (e.g., 1-column grid), ensuring no UI elements are cut off.
* **Global Standards:**
  + **Date Format:** **Strictly DD/MM/YYYY**. No other formats (e.g., MM/DD/YYYY or Jan 1st) are permitted in display or input fields.
  + **Theme Optimization:** All UI components, including text, borders, and glass backgrounds, must automatically verify **WCAG 2.1 Contrast Ratios** in both **Light Mode** and **Dark Mode**. Icons and text colors must invert intelligently (e.g., Dark Text on Light Glass, White Text on Dark Glass).
  + **Feedback Indicators:**
    - **Spinners:** Elegant, animated glass-ring spinners for general loading.
    - **Progress Bars:** Gradient-filled bars with percentage text for upload/export actions.
  + **Error Handling Standard:**
    - **Visible Error Codes:** Every error screen, modal, or toast notification must display a unique, specific **Error Code** (e.g., ERR\_NET\_503, AUTH\_TOKEN\_EXP, DB\_WRITE\_FAIL) alongside the user-friendly message. This ensures IT support can pinpoint issues rapidly.

### 8.1. Guest / Landing Page

* **Purpose:** The initial splash screen displayed when the Electron application launches. Since the app is internal, this acts as the "Welcome" kiosk screen.
* **Visual Style:**
  + **Background:** A rich, moving gradient mesh (deep blues/purples for Dark Mode, soft pastels for Light Mode) or a high-quality blurred photo of the college library.
  + **Container:** A single, large, centered "Frosted Glass" panel containing all content.
* **Key Elements:**
  1. **Header:**
     + College Logo (Top Center).
     + Language Switcher (Top Right): Toggle between "English" and "Kannada".
     + Theme Toggle (Top Right): Sun/Moon icon.
  2. **Hero Section:**
     + **Title:** "Central Library Management System".
     + **Subtitle:** "Empowering Students & Faculty with Infinite Knowledge."
  3. **About Section:**
     + A brief paragraph describing the library's capacity, digital resources, and operating hours.
  4. **Primary Action:**
     + **"Login" Button:** A prominent, pill-shaped glass button with a white border and inner glow. Hovering creates a "shine" effect.
       - *Action:* Navigates to the Login Page (Overlay or separate screen).
  5. **Footer:**
     + Version Number (e.g., v1.0.0).
     + Support Contact Email.

### 8.2. Login Page

* **Purpose:** Secure entry point for Staff and Administrators.
* **Visual Style:**
  + **Card:** A compact, centered Glassmorphism card (blurred background, rounded corners) floating above the main gradient background.
  + **Inputs:** Transparent input fields with thin white borders. On focus, the border glows (blue/gold depending on theme).
* **Key Elements:**
  1. **Header:** "Staff Login" / "ಸಿಬ್ಬಂದಿ ಲಾಗಿನ್" (bilingual support).
  2. **Form Fields:**
     + **Email:** Text input with icon (envelope).
     + **Password:** Masked text input with icon (lock) and a "Show/Hide" toggle eye icon.
  3. **Actions:**
     + **Login Button:** Primary glass button, full width. Triggers authentication.
     + **Forgot Password:** A subtle text link (underlined on hover) aligned to the right or bottom. *Action:* Redirects to the Forgot Password Page.
  4. **Feedback:**
     + Error messages (e.g., "Invalid Credentials") appear as a soft red glow around the input fields or a small notification banner within the card.

### 8.3. Forgot Password Page

* **Purpose:** Account recovery via email verification.
* **Visual Style:** Maintains the central Glassmorphism card layout but focuses on a multi-step process.
* **Workflow & Elements:**
  + **Stage 1: Identification**
    - **Instruction:** "Enter your registered email address to receive an OTP."
    - **Input:** Email Field.
    - **Action:** "Send OTP" Button.
  + **Stage 2: Verification (OTP)**
    - *Appears after successful email submission.*
    - **Input:** 6-digit OTP entry (individual glass boxes for each digit).
    - **Timer:** "Resend OTP in 30s" countdown.
    - **Action:** "Verify" Button.
  + **Stage 3: Reset**
  + *Appears after valid OTP.*
  + **Inputs:** "New Password" and "Confirm Password".
  + **Action:** "Update Password" Button.
  + **Success State:** A success animation (check mark) followed by auto-redirection back to the Login Page.

### 8.4. Dashboard Page

* **Purpose:** The central command center providing a real-time overview of library operations, analytics, and alerts.
* **Visual Style:**
  + **Grid Layout:** A responsive masonry or grid layout (CSS Grid) that adjusts columns based on screen width (Auto-sizing).
  + **Glass Panels:** Each widget (Cards, Charts, Logs) is contained within a distinct frosted glass panel.

#### Key Elements: A. KPI Summary Cards (Top Section)

* + **Visuals:** Vibrant, gradient-colored glass cards (e.g., Ocean Blue for Books, Sunset Orange for Fines, Emerald Green for Circulation) to distinguish metrics instantly.
  + **Interaction:** **Clickable**. Clicking a card navigates to the detailed list view for that specific metric.
  + **Metrics:**
    1. **Total Books (Titles):** *Action:* Redirects to Master Book Catalog.
    2. **Total Students:** *Action:* Redirects to Student Directory.
    3. **Books Issued Today:** *Action:* Redirects to Daily Circulation Report.
    4. **Overdue Books:** *Action:* Redirects to Overdue Management/Alerts Screen.
    5. **Total Fines Collected (Month):** *Action:* Redirects to Financial Report.
    6. **Lost / Damaged Books:** *Action:* Redirects to Stock Audit Report.

B. Inventory Analytics (Middle Section)

#### Style: Semi-transparent chart containers. Hovering over data points highlights the value using tooltips.

* + **Widgets:**
    1. **Books by Department:** *Pie/Doughnut Chart*. Visualizes the distribution of the library collection across different departments.
    2. **Students by Department:** *Bar Chart*. Shows active borrower density per department.
    3. **Most Issued Books:** *Horizontal Bar Chart* or *Top 5 List*. Highlights trending resources to aid in procurement decisions.

C. Audit & Compliance (Bottom Section)

#### Style: Scrollable list panels with timestamped entries and status indicators.

* + **Widgets:**
    1. **Recent Admin Actions:** Log of configuration changes (e.g., "Changed Fine Rate", "Added New Staff").
    2. **Staff Activity Log:** Tracking operational actions (e.g., "Staff A issued Book X", "Staff B collected Fine").
    3. **Data Import History:** Status logs of bulk CSV uploads (e.g., "Batch 2025 Students Import - Success").
    4. **Security Alerts:** Critical notifications (e.g., "Multiple failed login attempts", "Unauthorized access detected").

### 8.5. Notification Broadcaster Page

* **Purpose:** A dedicated interface for Staff/Admins to compose and send bulk communications.
* **Visual Style:**
  + **Split Layout:** A "Composer" glass panel on the left (60% width) and a "History/Preview" glass panel on the right (40% width).

#### Key Elements: A. Composer Panel (Left)

* 1. **Header:** "Broadcast Message" / "ಸಂದೇಶ ಪ್ರಸಾರ".
  2. **Targeting Section:**
     + **Recipient Type (Dropdown):**
       - *Options:* "All Students", "Specific Department", "Specific Students", "Overdue Students", "Issued Students".
     + **Dynamic Filters:**
       - *If Department selected:* Multi-select dropdown showing Dept Names.
       - *If Specific Student selected:* Auto-complete search bar (Input Register No/Name).
  3. **Message Details:**
     + **Subject Line:** Text input field.
     + **Message Body:** WYSIWYG Editor (Glass-styled toolbar) allowing text formatting.
  4. **Action Buttons:**
     + **"Send Broadcast":** Primary button (Green/Blue gradient). Triggers a confirmation modal summarizing the recipient count (e.g., "Sending to 150 Students?").
     + **"Clear":** Secondary button to reset form.

B. History & Status Panel (Right)

#### Header: "Recent Broadcasts".

* 1. **List View:**
     + Cards showing: *Date Sent*, *Subject*, *Target Group*, and *Status* (Sent/Failed).
  2. **Status Indicators:**
     + Green Dot: Successfully sent.
     + Red Dot: Failed (with error log tooltip).
     + Yellow Dot: Sending in progress.

### 8.6. Catalog Management Page

* **Purpose:** The core interface for managing the library's book inventory, including addition, editing, and bulk operations.
* **Visual Style:** Clean, data-heavy Glassmorphism interface. High readability is prioritized over decorative elements.

#### Key Elements: A. Top Control Bar

* 1. **Search & Filter:**
     + **Search Bar:** Large, centered input (search by Title, Author, or ISBN).
     + **Filters:** Dropdown for **Department** (e.g., "Computer Science").
     + **Sort:** Options for *Recently Added*, *Title (A-Z)*, and *Availability*.
  2. **Actions:**
     + **"Export Data" Button:** Triggers **Export Configuration Modal**.
       - **Scope Selection:**
         * **All Books** (Complete Database).
         * **Selected Rows** (Only items currently checked).
         * **Filtered View** (Matches current Department/Search criteria).
       - **Format:** **CSV** | **PDF**.
     + **"Import CSV" Button:** (Icon: Upload).
     + **"Add Book Manually" Button:** Primary Action (Icon: Plus).

B. Add Book Workflows (Modals)**1. Manual Add Modal**

#### Logic:

* + - **ISBN Field:** On focus loss (blur), the system triggers an API call (Google Books / Open Library).
      * *Success:* Auto-populates Title, Author, Publisher, Cover URL.
      * *Failure/Empty:* If left blank, system auto-generates a custom ID with prefix (e.g., AG-2025001).
  + **Form Fields:**
    - **ISBN:** (PK, Unique, Not Null).
    - **Title:** (Not Null).
    - **Author:** (Text).
    - **Publisher:** (Text).
    - **Department:** (Dropdown Select, Not Null).
    - **Price:** (Number, Optional).
    - **Quantity:** (Number, Not Null) -> Auto-creates this many "Book Copy" records.
    - **Cover Image URL:** (String).
    - **E-Book Link:** (String, Optional).
  + **Validation:** Highlight missing required fields in red.

**2. Bulk Upload (CSV) Modal**

* + **Step 1: Upload:**
    - Drag & Drop zone.
    - **Constraints:** Max **1000** records per batch. File size limit displayed.
    - **Helper:** "Download Sample CSV Format" link.
  + **Step 2: Preview & Validation (Editable Grid):**
    - Displays data in an Excel-like editable grid (Glass table).
    - **Auto-Fetch:** System attempts background fetch for rows with valid ISBNs but missing titles.
    - **Loading State:** **Live Progress Bar** ("Fetching metadata: 45%").
    - **Error Handling:** Rows with errors (e.g., duplicate ISBN, missing Title) are highlighted Red. Specific error messages appear in a "Status" column.
    - **User Action:** User can fix typos directly in the grid.
  + **Step 3: Confirmation:**
    - "Importing X valid books. Y rows skipped." -> Confirm Button.
    - **Loading State:** **Live Progress Bar** ("Writing to Database: 12%... 100%").

C. Book List (Main Content)

#### Layout: Infinite Scrolling Grid or List view.

* + **Bulk Actions:**
    - Checkbox selection (Single, Multiple, or "Select All").
    - **Delete Button:** Floats into view when items are selected. (Requires "Type CONFIRM" modal).
  + **Book Card Design:**
    - **Image:** Thumbnail of Cover URL. If invalid/null, show a stylized **Graphic Placeholder** (e.g., Book Icon with Dept Color).
    - **Metadata:**
      * **Title:** Bold, truncated after 2 lines.
      * **Sub-text:** ISBN | Author.
      * **Department:** Colored Badge (e.g., Blue for CSE).
    - **Availability:** Progress bar or text: "Available: 3 / Total: 10".
    - **Action Buttons (Small):**
      * **Edit (Pencil):** Opens Edit Modal (Same as Add, but ISBN is disabled).
      * **Manage Copies (Layers Icon):** Opens Copies Modal.

D. Interaction Modals**1. Manage Copies Modal**

#### List: Displays a row for every physical copy of the book.

* + **Row Elements:**
    - **Accession Number:** Unique ID based on ISBN (e.g., 986985868565-001, 986985868565-002).
    - **Status Control:**
      * **If Issued:** Displays "Issued" tag (Read-only status; cannot be manually changed here).
      * **If Not Issued:** Dropdown menu with options: *Available*, *Damaged*, *Lost*, *Maintenance*.
  + **Logic:** Updating a copy's status to *Damaged* or *Lost* immediately recalculates and updates the "Available Copies" count in the Catalog.
  + **Actions:**
    - "Add More Copies": Button/Input to increment total quantity.

**2. Book Detail Modal (On Card Click)**

* + **Header:** Large Cover Image + Title + **Action Buttons**.
    - **Buttons:** **Edit Book** | **Manage Copies**.
  + **Info Tab:** Full bibliographic details (ISBN, Publisher, Price, etc.).
  + **Circulation Tab:**
    - **Current Holders:** List of Students currently holding this book.
    - *Columns:* Student Name, Issue Date, Due Date.

### 8.7. Student Management Page

* **Purpose:** Administration of student records, including enrollment, editing, and bulk import.
* **Visual Style:** Consistent with Catalog Management (Glassmorphism tables and cards).

#### Key Elements: A. Top Control Bar

* 1. **Search & Filter:**
     + **Search Bar:** Input field (Search by **Name** or **Register No**).
     + **Filters:** Dropdown for **Department** (e.g., "Civil Engineering") and **Semester** (1-6).
     + **Sort:** Options for *Name (A-Z)*, *Register No*, and *Semester*.
  2. **Actions:**
     + **"Export Data" Button:** Triggers **Export Configuration Modal**.
       - **Scope Selection:**
         * **All Students**.
         * **Selected Students** (Checkbox selection).
         * **By Department** (Dropdown select).
         * **By Semester** (Dropdown select).
       - **Format:** **CSV** | **PDF**.
       - **Constraint:** Must handle backend pagination to ensure full dataset export, not just visible rows.
     + **"Import Students" Button:** (Icon: Upload).
     + **"Add Student Manually" Button:** Primary Action (Icon: Plus).
     + **"End of Semester Promotion" Button:** (Special Global Action).
       - *See Workflow below in Section E.*

B. Add Student Workflows (Modals)*(Standard Add/Import logic remains same)*C. Student List (Main Content)

#### Layout: Infinite Scrolling List.

* + **Bulk Actions (On Selection):**
    - **Quick Edit Semester:** Batch increment/decrement semester for *selected* rows (Simple update, no validation).
    - **Activate/Deactivate Status:** Toggle account status (e.g., Block/Unblock).
    - **Delete Button:** Permanently remove records (Requires Confirmation).
  + **Student Row/Card Design:**
    - **Primary Info:** **Name** (Bold) | **Register No** (Subtext).
    - **Secondary Info:** **Department** Badge | **Semester** (e.g., "Sem 4").
    - **Status Indicator:** Green Dot (Active) / Red Dot (Blocked).
    - **Action Buttons:**
      * **Edit (Pencil):** Opens Edit Modal with fields pre-filled.

D. Student Detail Modal (On Row Click)

#### Header: Student Name + Register Number (No Profile Image/Avatar).

* + **Tabs:**
    1. **Profile Details:** Read-only view of Email, Phone, DOB, Address.
    2. **Circulation History:**
       - *Current Loans:* Books currently issued to this student.
       - *History:* Past returned books.
    3. **Fines:** Summary of paid and unpaid fines.

E. End of Semester Promotion Workflow (Global Action)

#### Purpose: Bulk promote students to the next semester while flagging those with library liabilities.

* + **Trigger:** Clicking "End of Semester Promotion" in Top Bar.
  + **Workflow:**
    1. **System Scan:** The system checks **All Students** proposed for promotion.
    2. **Liability Check:** Identifies students with **Active Loans** (Unreturned books) or **Unpaid Fines**.
    3. **Pre-Promotion Report (Defaulter List):**
       - **Modal Display:** "Found$$X$$  
         students with pending liabilities."
       - **Action:** **"Download Defaulter Report"** (PDF/CSV).
       - *Use Case:* This report lists Student Name, Reg No, Dept, and Pending Items/Fines. It is designed to be sent to **Department HODs** to withhold results or notify students.
    4. **Confirm Promotion:**
       - **Action:** "Proceed with Promotion".
       - **Logic:** Increments semester count by +1 for all selected students (e.g., Sem 1 -> Sem 2). Note: Promoting students does *not* clear their dues; the debt remains attached to their profile.

### 8.8. Staff Management Page (Admin Only)

* **Purpose:** A restricted module for Administrators to manage library staff accounts, assign roles, and audit activities.
* **Visual Style:** High-security Administrative view.

#### Key Elements: A. Top Control Bar

* 1. **Search & Filter:**
     + **Search Bar:** Input field (Search by **Name** or **Email**).
     + **Filters:** Dropdown for **Designation** (e.g., "Librarian", "Assistant").
     + **Status Filter:** Active / Disabled.
  2. **Actions:**
     + **"Add New Staff" Button:** Primary Action (Icon: User Plus).

B. Add/Edit Staff Modal

#### Form Fields:

* + - **Full Name:** (Text, Not Null).
    - **Email:** (Email, Not Null, Unique) - *System sends initial setup link here.*
    - **Phone:** (String, Not Null).
    - **Designation:** (Dropdown: "Assistant Librarian", "Counter Staff", "Data Entry").
  + **Access Permissions (RBAC Checklist):**
    - *A set of toggle switches defining what this staff member can do:*
      * **Catalog Access:** (Add/Edit Books).
      * **Circulation Access:** (Issue/Return Books).
      * **Student Management:** (Add/Edit Students).
      * **Fine Management:** (Collect/Waive Fines).
      * **Reports:** (View Analytics).
  + **Security Actions (Edit Mode Only):**
    - **"Reset Password":** Sends a password reset email to the staff member.
    - **"Force Logout":** Invalidates current session tokens.

C. Staff List (Main Content)

#### Layout: Grid View of Glass Cards.

* + **Card Design:**
    - **Header:** Name + Designation Badge.
    - **Body:** Email + Phone Contact info.
    - **Footer (Permissions Summary):** Small icons representing enabled modules (e.g., Book icon for Catalog, User icon for Students).
    - **Status:** Toggle Switch (Active/Disabled). Disabling immediately blocks login access.
    - **Actions:** **Edit** (Pencil) | **Delete** (Trash - Soft Delete).

D. Staff Activity Log (Detail View)

#### *Accessible by clicking on a Staff Card.*

* + **Overview:** Profile details and last login timestamp.
  + **Audit Trail (Timeline):**
    - A chronological list of actions performed by this user.
    - *Example:*
      * 10:05 AM - Issued **"Intro to Algorithms"** to **Student 123**.
      * 11:20 AM - Added **5 New Books** to Catalog.
      * 02:15 PM - Collected **Fine (₹50)** from **Student 456**.

### 8.9. Admin Management Page (Super Admin Only)

* **Purpose:** High-level management of system administrators. *Note: Access is strictly restricted to the root Super Admin or Admins with specific privilege levels.*
* **Visual Style:** High-security Administrative view with distinct styling (e.g., Gold/Shield icons) to differentiate from Staff management.

#### Key Elements: A. Top Control Bar

* 1. **Search & Filter:**
     + **Search Bar:** Input field (Search by **Name** or **Email**).
     + **Status Filter:** Active / Disabled.
  2. **Actions:**
     + **"Create New Admin" Button:** Primary Action (Icon: Shield Plus).

B. Create/Edit Admin Modal

#### Form Fields:

* + - **Full Name:** (Text, Not Null).
    - **Email:** (Email, Not Null, Unique) - *System sends initial setup link here.*
    - **Phone:** (String, Not Null).
  + **Security Actions:**
    - **"Reset Password":** Sends a secure password reset link.
    - **"Revoke Access":** Immediately disables the admin account.

C. Admin List

#### Layout: Grid View of Glass Cards.

* + **Card Design:**
    - **Header:** Name + "Admin" Badge (Gold Color).
    - **Body:** Email + Phone Contact info.
    - **Status:** Toggle Switch (Active/Disabled). Disabling immediately blocks system-wide control.
    - **Actions:** **Edit** (Pencil) | **Delete** (Trash). *Note: Root Admin cannot be deleted.*

D. Admin Activity Log (System Audit Trail)

#### *Accessible by clicking on an Admin Card.*

* + **Audit Scope:** Tracks critical, high-impact actions affecting the entire system.
  + **Log Examples:**
    - Jan 14, 10:00 AM - **Updated Global Fine Policy** (Increased daily rate to ₹5).
    - Jan 13, 04:30 PM - **Created New Staff Account** ("John Doe").
    - Jan 12, 09:15 AM - **Exported Full Student Database** (Security Alert).
    - Jan 11, 02:00 PM - **Purged Inactive Student Records** (Batch Action).

### 8.10. Department Management Page (Admin Only)

* **Purpose:** Management of the master list of academic departments. This data populates dropdowns used in Student Registration and Book Cataloging.
* **Visual Style:** Master Data Management view (Clean, list-focused).

#### Key Elements: A. Top Control Bar

* 1. **Search:**
     + **Search Bar:** Input field (Search by **Name** or **Code**).
  2. **Actions:**
     + **"Add Department" Button:** Primary Action (Icon: Building Plus).

B. Add/Edit Department Modal

#### Form Fields:

* + - **Department Name:** (Text, Not Null, Unique, e.g., "Computer Science Engineering").
    - **Department Code:** (Text, Not Null, Unique, e.g., "CSE").
    - **Description:** (TextArea, Optional).
  + **Validation:**
    - Real-time check to prevent duplicate Codes or Names.

C. Department List (Main Content)

#### Layout: Responsive Glass Table or Card Grid.

* + **Row/Card Data:**
    - **Primary:** Dept Name (Bold) | Code (Badge).
    - **Statistics (Live Counts):**
      * *Linked Books:* (e.g., 1,250 Titles).
      * *Linked Students:* (e.g., 450 Active).
  + **Actions:**
    - **Edit (Pencil):** Update Name or Description.
    - **Delete (Trash):**
      * *Constraint:* **Disabled/Greyed out** if Linked Books > 0 or Linked Students > 0.
      * *Tooltip:* "Cannot delete department with active resources. Reassign or delete them first."
      * *Force Action:* If 0 links, prompts "Are you sure? This action is irreversible."

### 8.11. Fine Management & History Page

* **Purpose:** Management of financial dues, collecting pending fines, and audit trail of receipts.
* **Visual Style:** Data-dense Ledger view (Tabular) with Tabs.

#### Key Elements: A. Tabs (Top Bar)

* + **Tabs:** **Pending Dues / Uncollected Fines** | **Collection History (Ledger)**.
  + **Global Actions:**
    - **"Export Ledger":** Triggers **Export Configuration Modal**.
      * **Scope:**
        + **Current View** (Based on filters).
        + **All Pending Dues**.
        + **Collection History by Date Range**.
        + **By Collected Staff** (Admin Only).
      * **Format:** **CSV** | **PDF**.

B. Tab 1: Pending Dues (Uncollected Fines)

#### Purpose: List of students with outstanding balances waiting to be cleared.

* + **Search:** Search by Student Name/Reg No.
  + **List View Columns:**
    - **Student Info:** Name | Reg No.
    - **Total Dues:** **₹150.00** (Red Text).
    - **Breakdown:** "Damage: *Book Title* (₹100) + Overdue (₹50)".
    - **Actions:** **Collect / Settle** Button.
  + **Settlement Modal (On Click):**
    - **Editable List:** Shows individual fine entries.
    - **Edit Capability:** Staff can click the **Amount** or **Reason** to edit them manually (e.g., reducing a fine amount based on discretion).
    - **Action:** **"Mark as Paid"** (Generates Receipt) or **"Save Changes"**.

C. Tab 2: Fine History Table (Ledger)

#### Layout: Sortable Glass Table.

* + **Columns:**
    - **Receipt No:** (Clickable link to view details).
    - **Date:** Timestamp (e.g., "14/01/2026 10:30 AM").
    - **Student:** Name | Register No.
    - **Reason/Context:** e.g., "Overdue: *Introduction to Algorithms* (5 Days late)".
    - **Amount:** Currency formatted (e.g., ₹25.00).
    - **Collected By:** Staff Name.
  + **Row Actions:**
    - **Print Icon:** Immediately triggers browser print dialog for the specific receipt.
    - **Download Icon:** Generates and downloads a PDF receipt.

D. Receipt Preview Modal

#### Trigger: Clicking on a Receipt Number or the "View" action.

* + **Visual Style:**
    - **Professional Layout:** Designed to look like a high-end official document.
    - **Theme Integration:**
      * **Light Mode:** Crisp white paper background, dark serif typography, subtle college watermark.
      * **Dark Mode:** Deep slate background, off-white typography, gold accent borders (screen view only; prints always default to high-contrast white).
    - **Branding:** High-resolution College Logo header.
  + **Content Layout:**
    - **Header:** College Logo + Library Address.
    - **Title:** "OFFICIAL FINE RECEIPT".
    - **Meta Data:** Receipt No: F-2026-0001 | Date: 14/01/2026.
    - **Payer Details:** Received with thanks from **Student Name** (Reg: **ID**).
    - **Payment Details (Table):**
      * *Item:* Overdue Fine - **Book Title**.
      * *Calculation:* 5 Days @ ₹5/day.
      * *Total:* **₹25.00**.
    - **Footer:**
      * "Collected by: **Staff Name**".
      * "System Generated Receipt - Signature not required."
  + **Actions:**
    - **Print**: Print to thermal printer or A4.
    - **Download PDF**: Save locally.
    - **Close**: Dismiss modal.

### 8.12. System Audit Logs Page (Super Admin Only)

* **Purpose:** A centralized, immutable record of every significant action taken within the system for security compliance, troubleshooting, and accountability.
* **Visual Style:** Terminal/Log file aesthetic (Monospaced fonts for data) within a Glass container.

#### Key Elements: A. Advanced Filters (Top Bar)

* 1. **Date Range:** Start Date / End Date Picker (DD/MM/YYYY).
  2. **Module Filter:** Dropdown (e.g., "Catalog", "Circulation", "Security", "Settings").
  3. **Action Type:** Dropdown (Create, Update, Delete, Export).
  4. **Search:** Free text search (Search by Actor Name, IP, or Description).

B. Log Table (Main Content)

#### Layout: High-density list view.

* + **Columns:**
    - **Timestamp:** DD/MM/YYYY HH:mm:ss.
    - **Actor:** Name (Role Badge: Admin/Staff).
    - **Action:** Colored Badge (Green=CREATE, Blue=UPDATE, Red=DELETE, Yellow=SECURITY).
    - **Module:** Text.
    - **Description:** Summary of the event.
    - **IP Address:** Source IP.
  + **Interaction (Diff View):**
    - Clicking a row (specifically for UPDATE actions) expands a panel showing the **Metadata** (Before vs. After values).
    - *Example:* "Fine Rate changed: **₹2.00** -> **₹5.00**".

C. Export Tools

#### Action: "Export Logs" Button.

* + **Configuration Modal:**
    - **Scope:**
      * **Full Audit Trail** (All time).
      * **Current Filtered View** (Matches Date/Module/Action filters).
      * **By Actor** (Select Staff/Admin).
    - **Format:**
      * **Encrypted ZIP** (JSON/CSV) - For official audits.
      * **PDF** - For printable reports.
  + **Use Case:** Required for annual college audits or security incident investigations.

### 8.13. Library Rules & Policy Management (Admin Only)

* **Purpose:** The central configuration panel where Admins define the logic governing circulation, fines, and static content. Changes here globally affect system behavior and are versioned for audit purposes.
* **Visual Style:** Tabbed Settings Panel (Clean forms with explanatory tooltips).

#### Key Elements: A. Tab 1: Borrowing Rules (Profiles)

* + **Profile Selector:** Tabs for$$Active Students$$  
    |$$Alumni$$  
    |$$Staff$$  
    .
  + **Configuration Fields (Per Profile):**
    - **Max Books Allowed:** Number Input.
    - **Loan Period:** Number Input (Days).
    - **Max Renewals Allowed:** Number Input.
    - **Grace Period:** Number (Days before fine calculation starts).
    - **Auto-Renew Enabled:** Toggle (Specific to Staff).
  + **Blocking Logic (Student/Alumni Only):**
    - **Block Borrowing If:**
      * Unpaid Fines exceed ₹$$Input$$  
        .
      * Account Status = **Blocked**.

B. Tab 2: Fine & Financial Settings

#### Base Calculations:

* + - **Daily Fine Rate (₹):** Currency Input (e.g., ₹5.00).
    - **Max Fine Cap:**
      * Per Transaction (e.g., Max ₹500 per book).
      * Per Student (e.g., Total pending cannot exceed ₹2000).
  + **Damage Fine Presets (Customizable List):**
    - *Admin can Add/Edit/Delete specific damage types to speed up the Return workflow.*
    - **Grid:**
      * Torn Pages -> ₹50.00.
      * Water Damage -> ₹100.00.
      * Binding Repair -> ₹30.00.
      * Lost Book -> [Book Price] + [Processing Fee ₹50].
  + **Staff Permissions:**
    - **"Allow Staff to Edit Fine Amount":** Toggle (Yes/No).
    - **"Allow Staff to Waive Fines":** Toggle (Yes/No). *If Yes, system forces a "Reason" text input.*

C. Tab 3: Holiday & Calendar Settings

#### Weekly Schedule:

* + - **Weekly Holidays:** Checkboxes$$Mon$$  
      ...$$Sun$$  
      (Default: Sunday).
  + **Calculation Logic:**
    - **"Exclude Holidays from Fine Calculation":** Toggle (Yes/No). *If Yes, overdue fines skip Sundays/Holidays.*

D. Tab 4: General Rules (Static Content)

#### Purpose: Content management for the "Rules & Regulations" section displayed on the Guest/Landing Page.

* + **Editor:** WYSIWYG (Rich Text) Editor.
  + **Content:**
    - Silence must be maintained.
    - ID cards are mandatory.
    - Use of mobile phones is prohibited.
  + **Action:** "Publish to Guest Page".

E. Bottom Action Bar

#### "Reset to Defaults": Reverts to system presets.

* + **"Save & Publish Policy":**
    - *Security:* Requires **Admin Password** re-entry.
    - *Audit:* Increments policy\_version (v1.2 -> v1.3) and logs the specific changes in **Audit Log**.

### 8.14. Application Settings Page

* **Purpose:** Configuration of local application behavior, user personalization, and hardware integration. Accessible to all Staff/Admins, though specific tabs are role-restricted.
* **Visual Style:** Sidebar Navigation Layout (Left Sidebar for categories, Right Panel for settings).

#### Key Elements: A. Sidebar Categories

* 1. **Appearance & Feedback** (All Users)
  2. **Account Security** (All Users)
  3. **Hardware & Peripherals** (All Users)
  4. **Data & Maintenance** (Admin Only)
  5. **Email & Communication** (Admin Only)
  6. **System Security** (Admin Only)

B. Tab 1: Appearance & Feedback

#### Theme:

* + - **Mode:** Radio Buttons: **Light** | **Dark** | **System Default**.
    - **Glass Intensity:** Slider to adjust the blur strength of UI panels.
  + **Sound Feedback:**
    - **Success Beep:** Toggle (Play sound on successful Scan/Save).
    - **Error Beep:** Toggle (Play distinct sound on Invalid Barcode/Error).
  + **Language (Locales):**
    - **Interface Language:** Dropdown: **English (Primary)** | **Kannada**.
    - *Note:* Changing language requires an app restart.
  + **Accessibility:**
    - **Font Scaling:** Slider (80% to 150%).
    - **High Contrast:** Toggle Switch (For visually impaired users).

C. Tab 2: Account Security

#### Change Password:

* + - **Current Password:** Masked Input.
    - **New Password:** Masked Input (Strength meter validation).
    - **Confirm New Password:** Masked Input.
  + **Session Settings:**
    - **Auto-Lock Timer:** Dropdown (e.g., "After 5 mins of inactivity").

D. Tab 3: Hardware & Peripherals

#### Barcode Scanner:

* + - **Input Mode:** Dropdown: **Keyboard Emulation (Default)** | **Serial Port**.
    - **Prefix/Suffix:** Settings to strip characters if the physical scanner adds them automatically.
  + **Receipt Printer:**
    - **Default Printer:** System Dropdown (Select Thermal Printer).
    - **Paper Size:** **58mm** | **80mm** | **A4**.
    - **Auto-Print:** Toggle (Automatically print receipt after fine collection).

E. Tab 4: Data & Maintenance (Admin Only)

#### Cloud Database Configuration:

* + - **Connection URI:** Input Field (e.g., mongodb+srv://admin:pass@cluster0...).
    - **Action:** **"Test Connection"** (Validates reachability).
    - **Action:** **"Save & Restart"** (Applies changes).
  + **Backup Operations:**
    - **"Create Local Backup":** Dumps the MongoDB database to a local encrypted JSON file.
    - **"Restore from Backup":** Upload a backup file to overwrite the current database (Requires Super Admin Password).
  + **Backup Automation:**
    - **Auto-Backup:** Toggle (Enable/Disable).
    - **Frequency:** Dropdown$$Daily$$  
      |$$Weekly$$  
      .
    - **Manual Actions:** **"Create Local Backup"** | **"Restore from Backup"** (Requires Admin Password).
  + **Cache Control:**
    - **"Clear App Cache":** Wipes temporary UI states (useful for troubleshooting display bugs).
  + **Factory Reset:**
    - **"Reset System":** (Danger Zone). Wipes all catalog and user data. *Requires 2-step verification (Password + OTP).*

F. Tab 5: Email & Communication (Admin Only)

#### Master Control:

* + - **Email Service Status:** Toggle Switch **Enable / Disable**. *Turning this off suppresses all outgoing emails instantly.*
  + **Service Provider Configuration:**
    - **Provider Type:** Dropdown **SMTP Server** | **SendGrid API** | **AWS SES**.
    - **Conditional Fields (Based on selection):**
      * *If SMTP:* Host, Port, Secure (SSL/TLS), Username, Password (Masked).
      * *If Cloud API:* API Key (Masked), Region.
    - **Sender Identity:**
      * **"From" Name:** Input (e.g., "College Library").
      * **"From" Email:** Input (e.g., "library@college.edu").
  + **Connectivity Test:**
    - **Action:** **"Send Test Email"** Button.
    - **Logic:** Sends a "Hello World" ping to the currently logged-in Admin's email to verify credentials.
  + **Event Logic Toggles:**
    - *Granular control over which actions trigger emails:*
      * **Issue Receipt:** Toggle.
      * **Return Receipt:** Toggle.
      * **Renewal Confirmation:** Toggle.
      * **Broadcast Messages:** Toggle.
      * **Overdue Alerts (Cron):** Toggle.
      * **Fine Payment Receipt:** Toggle.
  + **Action:** **"Save Configuration"** (Logs to Audit Trail).

G. Tab 6: System Security (Admin Only)

#### Login Security:

* + - **Max Failed Login Attempts:** Number Input (Default: 5). *Account locks after limit.*
  + **Password Complexity Rules:**
    - **Min Length:** Number Input (Default: 8).
    - **Require Uppercase:** Toggle.
    - **Require Number:** Toggle.
    - **Require Symbol:** Toggle.
  + **Critical Action Protection:**
    - *Require Admin Password Re-entry for:*
      * $$x$$  
        **Bulk Delete Operations**.
      * $$x$$  
        **Restore Backup**.
      * $$x$$  
        **Policy/Rule Changes**.
      * $$x$$  
        **Staff Role Modification**.

### 8.15. Circulation Desk Page (Staff & Admin)

* **Purpose:** The high-traffic operational hub for issuing and returning books. Designed for speed, keyboard accessibility, and minimal clicks.
* **Visual Style:** Split-View or Tabbed Interface with large input fields and high-contrast status indicators.

#### Key Elements: A. Mode Switcher & Session Context

* + **Tabs:** **ISSUE (F1)** | **RETURN (F2)** | **RENEW (F3)**.
  + **Status Banner:** Shows current terminal connectivity and scanner status.
  + **Session ID:** Hidden but logged TXN-YYYYMMDD-XXXX generated at the start of every interaction. Used for audit trails, dispute resolution, and staff accountability.

B. Mode 1: Issue Book (Check-Out)

#### Layout: Asymmetric Split.

* + - **Left Panel (70%):** Browse & Search Area (Switches between Student List and Book List).
    - **Right Panel (30%):** Transaction Cart & Summary.
  + **Workflow:**
    1. **Step 1: Identify & Validate Borrower (Left Panel)**
       - **Input:** Scan ID / Search Name.
       - **Smart Borrower Validation (Pre-Scan Check):**
         * *Status Check:* Is Student Account **Active**?
         * *Fiscal Check:* Are Total Unpaid Fines **below** the Block Threshold?
         * *Quota Check:* Is (Current Loans + Cart Items) **<** Max Borrow Limit?
       - **Result:**
         * *Pass:* Automatically focus the Book Input field.
         * *Fail:* **Block Interaction** and show Red Alert (e.g., "Borrow Limit Reached" or "Outstanding Fines > ₹500").
    2. **Step 2: Continuous Book Scanning (Left Panel)**
       - **Input Mode:** **Continuous Scanning** (Focus remains in input field).
       - **Workflow:**
         1. Staff scans ISBN/Accession Barcode.
         2. **Auto-Append:** System instantly validates and adds valid items to the **Item Cart** *without* manual confirmation clicks.
         3. **Debounce Logic:** Ignores identical scans within 2 seconds to prevent accidental double-reads.
         4. **Edge Case Handling:**

*Duplicate Scan:* If copy is already in cart -> **Toast Warning:** "Item already in cart".

*Reference Book:* If restricted -> **Toast Error:** "Reference item cannot be issued".

* + 1. **Step 3: Review & Confirm (Right Panel)**
       - **Item Cart:** Lists selected books.
       - **Action:** **"Confirm Issue"** Button (or Ctrl+Enter).
       - **Output:** Database updated -> **Automatic Email Receipt sent to Student.**

C. Mode 2: Return Book (Check-In)

#### Layout: User-Centric View.

* + **Workflow:**
    1. **Step 1: Identify Borrower**
       - **Input:** Scan/Type **Student Register No**.
    2. **Step 2: View Issued Items**
       - **Display:** List of books issued to student.
    3. **Step 3: Return & Quality Inspection**
       - **Action:** Click **"Return"** button.
       - **Quality Popup:** "Condition of Returned Book?"
         * Options: **Good** | **Damaged** | **Lost**.
       - **Damage Reporting (If Damaged Selected):**
         * **Presets (Checkboxes):** **Torn Pages** | **Water Damage** | **Binding Loose**.
         * **Staff Notes:** Optional text field (e.g., "Page 45-50 missing").
       - **Resolution Logic:**
         * Open **Resolution Modal** (Fine/Replacement/Both) as defined previously.
    4. **Step 4: Finalize & Dues**
       - **Action:** Proceed to Uncollected Fines if charges apply.
       - **Notification:** Email Receipt sent to Student.

### 8.16. Circulation Transaction Reports

* **Purpose:** A dedicated reporting module to view, filter, and export the complete history of book movements (Issues, Returns, Renewals) and regenerate past receipts.
* **Visual Style:** Data Grid with robust filtering options.

#### Key Elements: A. Top Control Bar

* 1. **Search & Filter:**
     + **Search:** Free text (Txn ID, Student Name, ISBN, Accession No).
     + **Date Range:** From/To Date Picker (DD/MM/YYYY format).
     + **Transaction Type:** Dropdown **All** | **Issue** | **Return** | **Renew**.
     + **Department:** Filter by Student Department.
  2. **Actions:**
     + **"Export Report" Button:** Triggers **Export Configuration Modal**.
       - **Scope Selection:**
         * **All Transactions** (Complete History).
         * **Filtered View** (Matches Search/Date/Type filters).
         * **By Department** (Specific Dept Reports).
         * **By Student** (Specific Student History).
       - **Format:**
         * **CSV** (Raw data for Excel analysis).
         * **PDF** (Formatted table document suitable for printing).

B. Transaction List (Main Content)

#### Layout: Sortable Glass Table.

* + **Columns:**
    - **Date/Time:** Timestamp (e.g., "15/01/2026 10:45 AM").
    - **Txn ID:** TXN-2026....
    - **Type:** Colored Badge (Green=ISSUE, Blue=RENEW, Grey=RETURN).
    - **Book:** Title | Accession No.
    - **Student:** Name | Register No.
    - **Staff:** Issued/Returned By.
    - **Status:** Current state (e.g., "Returned Late", "Active", "Overdue").
  + **Row Actions:**
    - **Print Receipt:** Icon (Regenerates the original thermal/A4 receipt).
    - **Download Receipt:** Icon (Saves the individual transaction receipt as PDF).

C. Receipt Templates (Regeneration Logic)

#### Issue Receipt: Lists books issued, assigned due dates (DD/MM/YYYY), and library policy summary.

* + **Return Receipt:** Lists books returned, any quality remarks (Damaged/Good), fines calculated, and clearance status.
  + **Renew Receipt:** Shows old due date vs. new extended due date.

### 8.17. System Error Pages

* **Purpose:** Fallback screens displayed when the application encounters critical failures (e.g., Network Loss, Server 500, 404 Not Found, Permission Denied).
* **Visual Style:** High-impact "Glassmorphism" overlay with a focus on troubleshooting information.

#### Key Elements: A. Central Error Container

* 1. **Visual Icon:** Large, animated vector illustration matching the theme (e.g., a disconnected plug for network issues, a confused robot for 404).
  2. **Error Code (Prominent):**
     + Displayed in large, monospaced font.
     + *Example:* CODE: ERR\_DB\_CONNECTION\_TIMEOUT
  3. **Human-Readable Message:**
     + "We couldn't connect to the Cloud Database."
  4. **Technical Details (Collapsible):**
     + A "Show Details" toggle that reveals the stack trace or specific API response for IT Admin debugging.
  5. **Actions:**
     + **"Retry Connection" Button:** (Primary) Re-initiates the failed request.
     + **"Return to Dashboard" Button:** (Secondary).
     + **"Report Issue" Button:** Auto-generates an email to support with the Error Code and Session ID attached.

### 8.18. System Health & Diagnostics Page (Admin Only)

* **Purpose:** A real-time dashboard for Administrators to monitor the operational status of all system components, connectivity, and hardware integrations.
* **Visual Style:** "Mission Control" aesthetic. Grid of status cards with prominent "Traffic Light" indicators (Green/Yellow/Red).

#### Key Elements: A. Status Indicators (Traffic Lights)

* + **Green (Operational):** Service is running within optimal parameters.
  + **Yellow (Degraded):** Service is running but with high latency or non-critical errors.
  + **Red (Critical):** Service is down or unreachable.

B. Core Component Cards

#### Cloud Database (MongoDB Atlas):

* + - **Status:** Connected / Disconnected.
    - **Metric:** Latency (e.g., "45ms").
    - **Action:** "Ping Test".
  1. **Local Native Store:**
     + **Status:** Healthy / Corrupt.
     + **Metric:** Sync Queue (e.g., "0 Pending Uploads").
     + **Action:** "Force Sync" (Pushes pending changes immediately).
  2. **API Server:**
     + **Status:** Online.
     + **Metric:** Uptime (e.g., "4d 12h").
  3. **Email Service:**
     + **Status:** Configured & Reachable.
     + **Provider:** SMTP / SendGrid.
     + **Action:** "Send Probe Email".

C. Peripheral Diagnostics

#### Barcode Scanner:

* + - **Status:** Detected / Not Found.
    - **Device Name:** (e.g., "Honeywell 1200g").
    - **Test Area:** A text input box to physically scan a barcode and verify input integrity.
  + **Receipt Printer:**
    - **Status:** Ready / Out of Paper / Offline.
    - **Action:** "Print Test Page".

D. Resource Usage (Electron App)

#### Memory: Usage graph (e.g., "250MB / 8GB").

* + **Disk:** Cache size (e.g., "15MB").
  + **Action:** "Clear Temp Files".

E. Actions Bar

#### "Run Full Diagnostic": Sequentially checks all services and generates a report.

* + **"Export Health Report":** Downloads a JSON/PDF file containing current system state and recent error logs (useful for sending to technical support).

## 9. Future Enhancements (Phase 2 Roadmap)

While the current system focuses on internal administration, the architecture is designed to support future student-facing extensions.

### 9.1. Student Web Portal (OPAC)

* **Concept:** A lightweight, read-only web application allowing students to check their status without visiting the library counter.
* **Architecture:**
  + **Type:** Separate React.js Single Page Application (SPA).
  + **Backend:** Consumes the same **MongoDB Atlas** database (via a restricted API layer).
* **Authentication:**
  + **Login:** Register Number + DOB (DDMMYYYY).
* **Proposed Features:**
  + **My Books:** View currently issued items and due dates.
  + **Fine Status:** View outstanding dues (Read-only).
  + **Catalog Search:** Browse available books and checking shelf location.
  + **Reservation:** (Optional) Place a hold on books currently issued to others.

(include, for possible thing give remark (text input) for audit logging. like in collect fine page and etc.., and all logos, images etc... should present in separate folder which will we add end moment and this not include book cover images...)